

# PATENT ABSTRACTS OF JAPAN

(11)Publication number : **08-288768**

(43)Date of publication of application : **01.11.1996**

(51)Int.Cl.

**H03G 3/02**

**G06F 15/80**

**H03H 17/00**

**H03H 17/02**

**H03H 17/02**

(21)Application number : **08-070423**

(71)Applicant : **SONY UNITED KINGDOM LTD**

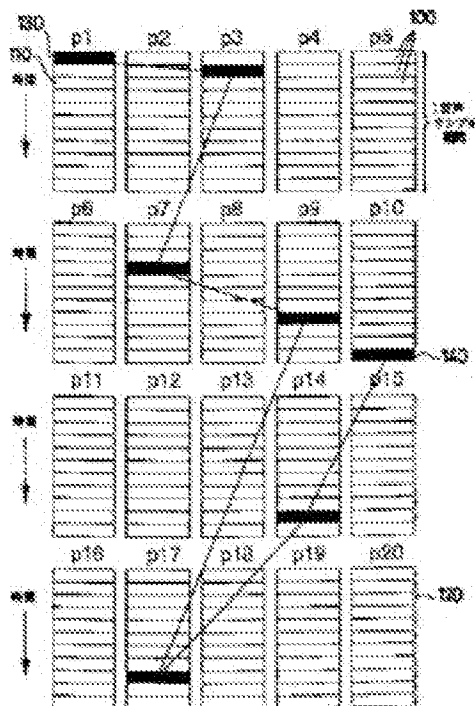
(22)Date of filing : **26.03.1996**

(72)Inventor : **MCCULLOCH CHRISTOPHER  
MICHAEL  
EASTTY PETER CHARLES  
KENTISH WILLIAM EDMUND  
CRANSTO**

(30)Priority

Priority number : **95 9506590** Priority date : **30.03.1995** Priority country : **GB**

## (54) **DIGITAL AUDIO PROCESSOR AND OBJECT CODE GENERATION METHOD FOR THE PROCESSOR**



(57)Abstract:

**PROBLEM TO BE SOLVED:** To efficiently allocate processing tasks to different processing units by parallelly allocating instructions to respective channels units by parallelly allocating instructions to respective channels inside the sub set of an audio channel to the plural audio processing units of a matrix array to be executed.

**SOLUTION:** In a signal processor for reacting to an audio mixing operation, the plural audio processing units P1-P20 or the like are arrayed in a matrix. In the respective units P1-P20, 11 pieces or the like of the instructions 100 are made processable corresponding to the respective channels inside the respective sub sets corresponding to the sub sets of the audio channel. Then, the units P1-P5, P6-P7, P11-P15 and P16-P20 perform parallel processings in successive audio sample periods, task instructions are parallelly and efficiently allocated to the required channels of the specified units and the tasks are parallelly executed.

(19)日本国特許庁 (J P)

(12) 公 開 特 許 公 報 (A)

(11)特許出願公開番号

特開平8-288768

(43)公開日 平成8年(1996)11月1日

(51)Int.Cl. <sup>8</sup>	識別記号	片内整理番号	F I	技術表示箇所
H 0 3 G 3/02			H 0 3 G 3/02	A
G 0 6 F 15/80			G 0 6 F 15/80	
H 0 3 H 17/00	6 1 1	8842-5 J	H 0 3 H 17/00	6 1 1 A
17/02	6 6 1	8842-5 J	17/02	6 6 1 E
	6 8 1	8842-5 J		6 8 1 G

審査請求 未請求 請求項の数8 OL (全 15 頁)

(21)出願番号 特願平8-70423

(22)出願日 平成8年(1996)3月26日

(31)優先権主張番号 9 5 0 6 5 9 0 : 0

(32)優先日 1995年3月30日

(33)優先権主張国 イギリス (G B)

(71)出願人 593081408

ソニー・ユナイテッド・キングダム・リミ  
テッド

Sony United Kingdom  
Limited

イギリス国 サリー, ウェブリッジ, ブ  
ルックランズ, ザ ハイツ (番地なし)

(72)発明者 クリストファ マイケル マッカラック

イギリス国 オックスフォードシャー, チ  
ッピング ノートン, スプリング ストリ  
ート 61

(74)代理人 弁理士 松隈 秀盛

最終頁に続く

(54)【発明の名称】 デジタル音声処理装置及び該装置のための目的コード生成方法

(57)【要約】

【課題】 複数の並列処理ユニットをもつデジタル音声処理装置において、異なる処理ユニットに効率よく処理タスクを配分すること。

【解決手段】 本発明は、多重プロセッサ・システムにおける目的コードの配分に関するもので、信号処理命令を並列信号処理ユニットの大きなアレイに効率よく割当てゐる種々の技法を開示している。

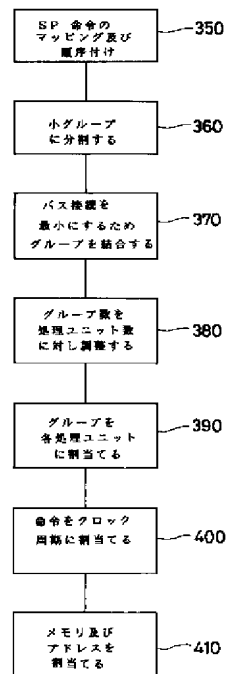


図3の処理アレイ用目的コードの作成と配分

**【特許請求の範囲】**

【請求項1】 複数の音声チャンネルに処理作用を行う複数の並列処理ユニットを有するデジタル音声処理装置であって、該音声チャンネルの少なくともサブセットについて、該サブセット内の各チャンネルに対する所要の処理を1より多い上記処理ユニットにより順次行い、上記処理ユニットの少なくとも1つが1より多い上記音声チャンネルに対する夫々の処理作用を行う、上記デジタル音声処理装置

【請求項2】 各処理ユニットは夫々所定の一連の繰返されるデータ処理命令を実行し、該命令の列は、上記音声チャンネル内の音声データの各音声サンプル期間内に一度実行される請求項1の装置。

【請求項3】 各処理ユニットに対する上記命令の列は、条件付き分岐命令を含まない請求項2の装置。

【請求項4】 相互接続された処理ユニットのアレイを有する多重プロセッサ・データ処理装置のための目的コード生成方法であって、

(i) 連続するデータ処理命令を含む原プログラムコードを生成するステップと、

(ii) 上記原プログラムコードを、上記処理ユニットのアレイ内の処理ユニットの数より多い数の、複数の命令グループに分割するステップと、

(iii) 上記命令グループの対の間の所要データ転送量を検出するステップと、

(iv) 上記グループの対を、検出された所要データ転送量が減少する順に並べるステップと、

(v) 上記グループの対を上記の並べた順に結合する場合、結合されたグループのサイズが各処理ユニットによって実行される命令の最大数を越えないとき、上記グループの対を上記並べた順に結合して結合されたグループを作り、該結合されたグループがグループ全体の合計所要データ転送量における最大の減少を与えるようにするステップとを含む、上記目的コード生成方法。

【請求項5】 更に

(vi) 結合されたグループの数が、使える処理ユニットの数より多いかどうかを検知するステップと、

(vii) 結合されたグループを、結合された各グループ内の命令の数の順に並べるステップと、

(viii) 命令の数が最も多いグループを命令の数が最も少ないグループと結合し、グループの数を、使える処理ユニットの数と等しいか又はそれより少なくなるように減らすステップとを含む請求項4の方法。

【請求項6】 連続するデータ処理命令を含む原プログラムコードを生成するステップと、  
原プログラムコード内の、単一の命令で置換えうる論理的に隣接する命令のグループを検出するステップと、  
検出された各命令グループを夫々単一の命令によって置換えるステップとを含む目的コード生成方法。

【請求項7】 検出された命令グループの各命令が2進

シフト命令であり、

当該命令グループに対する夫々の単一命令がビットシフト命令である請求項6の方法。

【請求項8】 検出された命令グループが加算命令と論理的に隣接する乗算命令を含み、

当該命令グループに対する夫々の単一命令が乗加算命令を含む請求項6又は7の方法。

**【発明の詳細な説明】**

【0001】

【発明の属する技術分野】本発明は、多重プロセッサ・システムにおけるオブジェクト（目的）コードの配分に関するものである。

【0002】

【従来の技術】並列に相互接続されたデータ処理デバイス又は処理ユニットのアレイを用いるデータ処理装置においては、異なる処理ユニットの間に処理タスクを配分することが必要である。これが（翻訳段の終わりに行われることが多い。）効率に行われるかどうかによって、該処理アレイの有用性及び動作効率が決定される。

【0003】このような多重並列データ処理装置の例として、音声ミキシング操作卓（コンソール）の如きデジタル音声（オーディオ）処理装置がある。以前に提案されたミキシング操作卓では、処理すべき幾つかの音声チャンネルの各々に夫々専用の処理ユニットを配分している。しかし、これは、種々の音声チャンネルの所要処理量が全く異なることがあるため、処理ユニットのアレイの使用効率を悪くしている。

【0004】

【発明が解決しようとする課題】したがって、本発明の主な課題は、複数の並列処理ユニットをもつデジタル音声処理装置において異なる処理ユニットに効率よく処理タスクを配分する方法を提供することである。

【0005】

【課題を解決するための手段】本発明は、複数の音声チャンネルに対して処理作用を行う複数の並列処理ユニットを含むデジタル音声処理装置を提供する。その装置は、該音声チャンネルの少なくともサブセット（小部分）について、該サブセット内の各チャンネルに対する所要の処理を1より多い処理ユニットで順次行い、処理ユニットの少なくとも1つが、1より多い音声チャンネルに対する夫々の処理作用を行うものである。

【0006】各処理ユニットは、夫々所定の一連の繰返されるデータ処理命令を実行し、その命令の列は、音声チャンネル内の音声データの各音声サンプル期間内に一度実行されるのが好ましい。また、各処理ユニットに対する命令の列は、条件付き分岐命令を含まないのがよい。そうすれば、各音声サンプル期間内に各処理ユニットにより実行されるプログラムは、他のどの音声サンプル期間内に実行されるものと同一であるので、これらの方策により、プロセッサ間の交信にバス調停が必要でな

いようにシステムを組立てることができる。バス転送は、バス転送が考えられる度毎に水平及び垂直バスの各々を1対(送信と受信)の処理ユニットに割当てること、で、予め解決できる。特定時にバスを使用する意図のない処理ユニットは、その時簡単にバス接続をトライステート(3状態)にすることができる。

【0007】本発明はまた、相互接続された処理ユニットのアレイをもつ多重プロセッサ・データ処理装置のための目的(実行)コード生成方法を提供する。その方法は、(i)連続するデータ処理命令を含む原プログラムコードを生成するステップ、(ii)原プログラムコードを、処理ユニットのアレイ内の処理ユニットの数より多い数の、複数の命令グループに分割するステップ、(iii)命令グループの対の間の所要データ転送量を検出するステップ、(iv)該グループの対を、検出された所要データ転送量が減少する順に並べるステップ、及び(v)上記グループの対を並べた順に結合する場合、結合されたグループのサイズが各処理ユニットによって実行される命令の最大数を越えなければ、上記グループの対を並べた順に結合して結合されたグループを作り、該結合されたグループがグループ全部の合計所要データ転送量における最大の減少を与えるようにするステップを含む。

【0008】上述の方法の結果、結合されたグループの数が処理ユニットの使用可能な(使える)数より多くなる場合、上述の方法は更に、(vi)結合されたグループの数が、使える処理ユニットの数より多いかどうかを検知するステップ、(vii)結合されたグループを、結合された各グループ内の命令の数の順に並べるステップ、及び(viii)命令の数が最も多いグループを命令の数が最も少ないグループと結合し、グループの数を、使える処理ユニットの数と等しいか又はそれより少なくなるように減らすステップを含む。

【0009】本発明はまた、次の如き目的コード生成方法を提供する。その方法は、連続するデータ処理命令を含む原プログラムコードを生成するステップ、原プログラムコード内の、単一の命令で置換えうる論理的に隣接する命令のグループを検出するステップ、及び検出された各命令グループを夫々単一の命令によって置換えるステップとを含む。

【0010】1つの好適な実施例では、検出された命令グループの各命令は2進シフト命令であり、当該命令グループに対する夫々の単一命令は、ビットシフト命令である。他の考えられる実施例では、検出された命令グループは加算命令と論理的に隣接する乗算命令を含み、当該命令グループに対する夫々の単一命令は乗加算命令を含む。

【0011】

【発明の実施の形態】以下、図面を参照して本発明を具体的に説明する。図1は、本発明を使用しうるデジタル

音声ミキシング操作卓を示す模式図である。図1において、ユーザは、制御パネル又はデスク10上の制御器を操作する。制御器は、スイッチ、フェーダ、可変抵抗器などである。制御パネル10はまた、例えば信号レベル、信号経路、等化器の動作などの表示も行う。パネル上の制御器及び表示デバイスとの情報交換は、パネルプロセッサ20によって処理される。

【0012】パネルプロセッサ20はまた制御プロセッサ30に接続され、制御プロセッサ30は、パネル上にユーザが設定した制御位置を示す情報をパネルプロセッサから受け、当該情報を用いて信号プロセッサ40の動作を制御する。

【0013】信号プロセッサ40は、入力/出力プロセッサ50からデジタル音声データを受け、制御プロセッサ30の制御の下に当該音声データを処理し、処理したデジタル音声データを入力/出力プロセッサ50に供給して出力させる。信号プロセッサ40は実際には、図3、4及び5を参照して後述する如き信号処理アレイとして具現される。

【0014】64~128チャンネルを処理する如き大きなミキシング操作卓では、種々の音声チャンネルを混合し、等化し、組合せを調整するのに大量の信号処理を必要とする。含まれる種々の原理を簡単に示すため、当該処理の極めて小部分のみを図2に示す。

【0015】図2に示すように、操作卓の1音声チャンネルに対するチャンネル処理小部分には、フェーダ(線形可変抵抗器)60、当該チャンネルの音声データを処理するためフェーダの位置を制御量(例えば利得のデシベル)に変換する位置変換器70、係数発生器80及び乗算器90が含まれる。

【0016】操作の際、ユーザは、フェーダ60を動かすことにより入力音声信号に適用すべき利得を指定することができる。フェーダの物理的位置は、デジタル化されて位置変換器70に送られる。位置変換器70は、フェーダ60のデジタル化された位置を対応する利得のデシベル(dB)値にマップし(割当て)、これを係数発生器80に送る。該係数発生器は、当該所要利得を乗算係数に変換する。入力音声データはそれから、乗算器90において当該乗算係数又は率により乗算される。

【0017】図1及び2を比べると分かるように、フェーダ60はパネル10の一部をなし、パネルプロセッサ20はフェーダのデジタル化された位置をサンプリングする。位置変換器70は制御プロセッサ30に含まれ、係数発生器80は制御プロセッサ30及び信号プロセッサ40によって具現される。最後に、実際は信号路の一部である図2のただ1つの部分、乗算器90は、信号プロセッサ40に含まれる。

【0018】前述したとおり、信号プロセッサ40は実際には、或る数の処理ユニットを含む信号処理アレイとして具現される。(制御プロセッサ30も同様に、制御

処理ユニットのアレイとして具体化できる。しかし、この具体例では、制御プロセッサ30は単一のマイクロプロセッサ・デバイスによって形成される。)。

【0019】図3は、信号プロセッサ40を形成する信号処理アレイの模式図である。該アレイを構成する個々の処理ユニットには、 $p1$ ,  $p2$ ,  $p3$ ,  $\dots$ ,  $p(n)$ ,  $\dots$ ,  $p(m)$  等々の番号が付けられている。これらは(少なくとも電氣的に)、処理ユニットの正方形又は方形のアレイとして配置される。例えば25個の処理ユニットのアレイでは、その電氣的配置は、夫々水平及び垂直の方向において $5 \times 5$ の処理ユニットの正方形アレイとなるであろう。かようなアレイの2つ以上をバスで一緒に連結できるであろう。

【0020】処理ユニットのアレイは、水平及び垂直方向の交信バスにより相互に接続される。図3では、処理ユニット $p1$ ,  $p2$ ,  $p3$ ,  $p4$ ,  $\dots$ は水平バスにより、処理ユニット $p1$ ,  $p(n)$ ,  $p(m)$ ,  $\dots$ は垂直バスにより相互接続されている。

【0021】水平及び垂直バスは、個々のバスに接続されたデバイスが、当該バスに接続された他のどのデバイスとも交信できるように配される。処理ユニットは、反復プログラムを各音声サンプル期間(44.1kHzのサンプリングレートの場合、約23ミリ秒)に1回ずつ繰返しながら実行する。条件付き分岐命令は使用されない。即ちプログラムが繰返される度に行われる処理動作は同一である(勿論、処理作用が施されるデータはサンプル期間毎に変化する。)

【0022】各処理ユニットによって各音声サンプル期間に実行されるプログラムは他のどの音声サンプル期間に実行されるプログラムとも同一であるから、バス調停(仲裁)を必要としないようにシステムを組立てることができる。バス転送は、バス転送が考えられる度毎に水平及び垂直バスの各々を1対(送信と受信)の処理ユニットに割当ててことで、予め解決できる。特定の時間にバスを使用する意図のない処理ユニットは、その時簡単にバス接続をトライステート(3状態)にすることができる。

【0023】各音声サンプル期間に各処理ユニットによって実行されるプログラムは、所定数のクロック周期を占めるが、これは、この具体例では512のクロック周期であり、44.1kHzのサンプル(サンプリング)レートの場合約23MHzのプロセッサ・クロック速度を与える。バス転送は、各クロック周期に行われるようにすることができる。

【0024】図4は、図3の処理アレイの動作を示す模式図である。図4は、図を分かり易くするため、処理ユニットを20個( $p1 \sim p20$ )しか示していないが、(前述の如く)実際にはもっと多くのユニットを使用することになる。

【0025】処理ユニットは図3に示したバス回路網で

連係されているので、個々のタスクの所要処理量を複数の処理ユニットに分配することができる。実際に、どの音声チャンネルもどの処理動作(等化段の動作など)も、特定の処理ユニットと結び付いてはいない。

【0026】図4は、特定の音声サンプル期間内の20個の処理ユニットの動作の例を示す。同図には、11個の命令100(垂直方向の区分で示す。)が処理ユニットによって当該音声サンプル期間内に実行される様子が示されている。これらの命令は同時に実行され、例えば、処理ユニット $p1$ に示された3番目の命令(命令110)は、処理ユニット $p2 \sim p20$ の各々に示された3番目の命令(例えば命令120)と同時に実行される。

【0027】処理タスクを複数の処理ユニットに配分するやり方の例を、図4では当該タスクを実行するのに使用される命令に陰影を付けて示す。この例では、処理タスクに要求される最初の処理は、処理ユニット $p1$ で命令130によって行われる。処理はそれから、バス回路網を通して処理ユニット $p3$ ,  $p7$ ,  $p9$ ,  $p17$ ,  $p14$ 及び $p10$ に順次転送され、処理ユニット $p10$ の命令140で終わっている。多くの他の所要処理タスクも、同様にアレイ内の全処理ユニットの間に差し挟まれる。

【0028】図5は、図3の処理アレイの動作の他の例を示す模式図である。図5は、個々の処理のタスクに必要な命令を、複数の処理ユニットのみならず複数の音声サンプル期間に配分する方法を示す。図5は、連続する音声サンプル期間 $n$ ,  $n+1$ ,  $n+2$ の間に5個の処理ユニット $p1 \sim p5$ が行う動作を示す。

【0029】図5に示す例では、処理は、処理ユニット $p1$ によって行われる命令150からスタートする。タスクは、 $p2$ によって続けられ、それから再び $p1$ に戻り、それから処理ユニット $p3$ に送られ、音声サンプル期間 $n+1$ において実行される。タスクは、 $p3$ から処理ユニット $p1$ 内の命令160に送られ、それから処理ユニット $p4$ 内の音声サンプル期間 $n+1$ における最後の命令に送られる。最後に、タスクは、第3の音声サンプル期間 $n+2$ における処理ユニット $p3$ 内の命令によって続けられ、音声サンプル期間 $n+2$ の終わり頃処理ユニット $p5$ によって行われる命令170で終了する。

【0030】図5に示した命令のチェーン(連鎖)は、各音声サンプル期間内のどの命令位置も再使用しないことに気付くであろう。実際には、1以上の音声サンプル期間だけ遅れるか又は進んだ類似の処理チェーンが、図5に示したチェーンとの間に挿入される。即ち、例えば、音声サンプル期間 $n$ における処理ユニット $p1$ 内の命令180は、音声サンプル期間 $n+1$ における同じ処理ユニット内の命令160と同一であるが、勿論、命令160により処理されるデータより1サンプル前の音声データについて行われる。

【0031】換言すれば、処理ユニット間でタスク配分を行うことにより、少なくとも音声チャンネルのサブセット（小部分）に対し、該サブセット内の各チャンネルに関連する所要の処理が1より多い処理ユニットによって順次行われ、少なくとも1つの処理ユニットが1より多い音声チャンネルに関連する反復処理動作を行うという特徴のおかげで、著しく効率がよくなる。

【0032】したがって、処理ユニット用の目的コードの作成には、命令を複数の処理ユニット、クロック周期及び音声サンプル期間に分配することが必要になる。また、バス交信やメモリ資源を複数の処理ユニットに割当てることも必要となる。

【0033】図6は、図1の操作卓内のプログラマブル処理デバイスのすべて（即ち、パネルプロセッサ20、制御プロセッサ30、信号プロセッサ40及び入力／出力プロセッサ50）に対する目的コードの作成を示す模式フローチャートである。これらの処理段は、汎用もしくは専用データ処理装置によって行われる。

【0034】適切な目的コードを作成する基本的技法は、次の文献に記載されている。

1. 1986年81回Audio Engineering Society(AES)大会予稿プリントに所載のW.Kentish 及びC.Bellによる「An automated approach to digital console design」、
2. 1986年81回AES大会予稿プリントに所載のP.Eastlyによる「Digital audio processing on a grand scale」、
3. 1986年81回AES大会予稿プリントに所載のC.McCulloch による「Automatic generation of microcode for a digital audio signal processor」。

【0035】上記の文献に記載された技法を要約すると、図2に示したものと形が類似の模式回路図（ただし、一般に大きさ及び複雑さは遙かに大きい。）をコンピュータ利用設計（CAD）システムにて作成する。該CAD模式図からネットリスト（netlist）を作り、これをコンパイル（翻訳）して図1の操作卓の種々のプロセッサに実行させるための目的コードを生成する。

【0036】したがって、図6は、図1の操作卓の信号及び制御処理のCAD模式図200からスタートする。前述のとおり、CAD模式図の一部分（フェーダ、位置変換器及び係数発生器など）は、制御プロセッサ30によって行われるタスクに関連し、CAD模式図200の他の部分は、信号処理アレイ40によって行われるタスクに関連している。

【0037】CAD模式図からネットリスト210を作成するが、これは、基本的には連係された数学的又はデータ処理の命令にCAD模式図を直接翻訳したものである。ネットリストの小部分の例をあとで述べる。データベース・コンパイラ220は、該ネットリストを処理して「データベース」230を生成する。該データベースは、コード生成目的のために（処理構造を「平坦に」表

示するため）ネットリスト内に存在するどんな階層構造も無視した（基本的なデータ処理命令の表現で）処理構造の全体を表すデータファイルである。しかし、その階層構造は、エラーメッセージを送って手直しをする目的のために存続される。

【0038】データベース230は信号プロセッサ（SP）コード発生器240によってアクセスされ、該SPコード発生器は、信号処理アレイの処理ユニットの処理特性を定義するデータ250と、処理ユニットの構成を定義するデータ260とを用いて目的コードを生成し（個々の処理ユニットの処理能力に適する命令のグループで）、これをそれからSPロード245により信号プロセッサ40にロード（転送）する。SPコード発生器240は、異なる処理ユニット、異なるクロック周期、異なるサンプル期間及び異なるバス転送期間に動作を割当てる。この部分の動作は、あとでもっと詳細に述べる。

【0039】制御コンパイラ270は、ミキシング操作卓のパネル10の構成を示すフロントパネル定義データ280と、各処理ユニットへのタスクの割当てを示す、SPコード発生器240からの割当てデータとを受け、制御プロセッサ30、パネルプロセッサ20及び入力／出力プロセッサ50によって実行されるべきコードを発生する。このコードは、リンカ／ロード290に送られる。該リンカ／ロードは、従来どおりの動作をするもので、ライブラリプログラム及びデータファイル300を受け、図1の操作卓の制御プロセッサ（並びにパネル及び入力／出力プロセッサ）に供給すべき目的コード310を生成する。

【0040】コンパイル（翻訳）された命令の異なる処理ユニットへの割当て（配分）について、これより図7を参照して説明する。図7は、SPコード発生器240によって行われる如き、図3の処理アレイ用の目的コードの作成及び配分を示す一層詳細な模式フローチャートである。

【0041】図7のステップ350で、初期コードを調べて余分な命令を除く。この命令のマッピング及び順序付け技法は、同じ結果を与える単一の複合命令に（簡単な所定のマッピング表で）翻訳できる、論理的に隣接する命令を探索することを含む。この技法は、あとで更に図8及び9を参照して説明する。この段階でまた、他の命令によって発生された変数を遅延させたものが必要となる命令を探索する。このタイプの命令を処理する仕方は、あとで図10及び11を参照して説明する。

【0042】ステップ360で、ネットリストを構成する命令を、あとで図9を参照して述べる小グループの如き、小さな任意のグループに分割する。ステップ370で、結合されたグループが単一の処理ユニットに適正な最大サイズ（例えば、512の命令）を越えないことを条件として、単一の処理ユニットに受入れられる命令の

グループを作る目的で、上記グループを一層大きいグループに合体する。

【0043】これを行うため、これらのグループを考えられるすべての対に組合せてテストし、考えられる各対の間の所要バス交信量を評価する。それらのグループ対をそれから、所要バス交信量が減少する順に並べる。この並べたリストの一番上から始めて、第1のグループ対（即ち、考えられる対の組合せのうち、該対間の所要バス交信量が最も多いもの）を結合する。上記リストの下の方に移り、各対を結合する。ただし、他のグループが前に2回以上結合され、それらが既に間接的に結合されているか、又は結合の結果グループのサイズが1処理ユニットの処理可能なサイズより大きくなることが検知されたときは、各対の結合は行わない。

【0044】例えば、5つのグループA、B、C、D及びEをもつ極めて簡単なシステムの例においてそれらのグループを考えられるすべての対に組合せて検討し、それらの対を所要バス交信量の順に並べる。その結果、次のような対の順になったとする。

AD（所要バス交信量が最も多い。）

DE

AE

BD

AB

BE

AC

BC

CD

CE（所要バス交信量が最も少ない。）

【0045】このリストを下方に移動し、幾つかのグループの対が結合される。ただし、結合されたグループが1処理ユニットの処理できるサイズを越えてはならない、との条件が常に付いている。最初にこのテストに合格すれば、まずグループAとDを結合する。それは、この対が、2グループ間で最も多い所要バス交信量をもつからである。次いで、グループDとE（第2の対）を結合する。しかし、第3の対（AとE）を検討すると、A、D及びEが前の2つの結合ステップで既に結合されていることが分かる。したがって、第3の対は、何も操作する必要がない。

【0046】この処理は、グループ対（実際には、何百もの多くの対になるであろう。）のリストの下方へと続けられる。

【0047】結合されたグループは、個々のグループと類似の形をしているが、それらより大きい。例えば、図9のグループを他の類似のグループと結合して論理的に相互連絡された命令の一層大きな、ただし類似のグループを発生することができるであろう。

【0048】所要バス交信量の評価について、次に簡単に述べる。例えば、第1のグループが例えば3つの入力

変数a、b及びcを要求し、3つの出力変数d、e及びfを発生したとすると、a～fのすべてを送信するのに1以上のバスが必要になるであろう。しかし、当該グループを、変数bを発生し変数eを要求した他のグループと組合せ、2つのグループを概念的なグループ対として一緒に配置できれば、もはや変数b及びeをバス回路網を介して送信する必要がなくなるので、該対の所要バス交信量はバスで交信される2つの入力変数と2つの出力変数になるであろう。

【0049】図7のステップ370の終わりで、グループの数は劇的に減少するが、それでもなお使用できる処理ユニットの数より多いことがある。よって、ステップ380で、所要グループ数を作るための最終調整が行われる。即ち、まずステップ370で出力されるグループをグループサイズの減少順に並べ、それから最小グループを最大グループと任意に結合して、最後に当該グループが1処理ユニットで処理できる命令数に達するようにし、それから続けて2番目に最も大きいグループを次に最も小さい残りのグループと結合する。以下同様である。この結果、使用可能な処理ユニットの数に等しいか又はその数より少ない幾つかのグループが得られる。

【0050】グループ中最大のものにグループ中最小のものを加えて結合したグループが、単一の処理ユニットで処理できる命令の数を越える場合、最小グループを原結合（ステップ370で行われる。）の位置で分割し、分割すべき特定の結合を、他（最大）のグループに加える分割されたグループが適正なサイズの部分を与えて、1処理ユニットの処理能力にできるだけ近づくように選択する。

【0051】次に、各命令グループを夫々の処理ユニットに割当てることが必要になる。これは、ステップ390で行われる。即ち、合計で最大の所要バス交信量をもつグループ（ステップ380で生成されたもの）を検出し、当該グループを処理ユニットの仮想無限アレイ上の任意の位置に割当てて。

【0052】次に最も多い所要バス交信量をもつグループを、第1グループとの交信が必要であると仮定して隣接する処理ユニットに割当てて。そうでない場合は、別個の仮想アレイ上に配置する。

【0053】ステップ390を続け、3番目に最も多い所要バス交信量をもつグループを、処理ユニットの仮想アレイ上の次の位置においてテストする。

—最初の2グループは互いに交信し、最初の2グループの少なくとも一方は第3のグループと交信すると仮定して、最初の2グループを接続するバスと同じバスに沿って、

—既に割当てられた2グループの第1のグループに接続された垂直バスに沿って（当該グループは第3グループと交信すると仮定して）、

—既に割当てられた2グループの第2のグループに接続

された垂直バスに沿って（当該グループは第3グループと交信すると仮定して）。

【0054】上記第3のグループが最初の2グループのどちらとも交信しない場合、それを、処理ユニットの3番目の仮想アレイ上に配置する。幾つかの仮想アレイが出来上がるに従い、これらを、それら（即ち、2以上の各仮想アレイ上のグループと交信するグループ）の間に共通のリンク（連係）が見付かる度に合体することができる。各テスト位置で、合計バス交信量を評価して、バス交信量が最もよく減少する位置を選択する。

【0055】この処理は、残りのグループに対して続けられる。即ち、各グループを順番に処理ユニットの仮想アレイ上の使用可能なすべての異なる位置でテストし、当該グループが交信する必要がある、既上記アレイ内にあるグループを検討する。この処理の終わりに、これらのグループを、互いに通信し合うグループ集団で、対応する数の処理ユニット位置に割当てて。これらの集団を、互いに隣接して配置する。この段階で、グループは、コードが実行されるべき物理的ハードウェアと同じ構成の仮想アレイ上に配置されていない。

【0056】したがって、アレイをそれから、仮想アレイ内の最も数の多いグループを対応する現実の処理ユニットに割当てるように調節する。仮想アレイ内の、処理ユニットの実際のアレイと「離れる」（即ち、仮想アレイ上に実際の処理ユニットに対応しない位置をもつ）グループはいずれも、次第に所要バス交信量の順に実際の処理ユニットのアレイ（前述した処理アレイ例の中の正方形アレイ）内の使用可能な位置に移し、最後に、すべてのグループを実際のアレイ内の夫々の処理ユニットに割当てることができる。これらの移された各グループの位置を、どのバスにも最低の最大交信量を与える（考えられる使用可能な位置内の）位置として（各グループに対し順番に）選択する。

【0057】ステップ400で、各処理ユニット内の命令を、当該処理ユニットによって実行されるクロック周期に割当てて。これは基本的に、それらの命令を他の命令への依存関係の連鎖の中に入れて結合することにより行われる。例えば、変数aを2つの他の変数b及びcから導出する場合、変数aを発生する命令は、変数b及びcを発生する命令の後に実行されねばならない。實際上パイプライン化されたプロセッサを使用する場合、変数aを発生する命令は、6命令パイプラインを使用するとすれば、変数b及びcの遅いものが発生されてから、例えば6命令後に行われねばならない。

【0058】変数aを発生する命令を実行するための時間上の他の制約は、変数b及びcの早いものの発生後1サンプル期間に、当該変数が新しい値によって重ね書きされる可能性があることである（処理命令は前述の如くサンプル期間毎に1回繰返されるので）。変数b及びcの前の値を、それが発生されてから1サンプル期間より

長く使うためにメモリ位置に記憶させることは可能であるが、これは資源の浪費になるので避ける方がよい。

【0059】最後にステップ410で、メモリ及びメモリアドレスを、1サンプル毎に記憶又は転送する必要のある変数に割当てて。図8は、CAD模式図200（図6）から生成されるネットリスト210の小部分を示す模式図である。

【0060】該ネットリストは、低水準（基本的）データ処理又は数学的命令で連係されたリストであり、図8に示す例は乗算及び加算命令である。具体的にいえば、第1の音声信号（音声1）が係数（係数1）によって乗算され、その積が第2の音声信号（音声2）及び他の係数（係数2）の積に加算される。これら2つの積の和に更に他の係数（係数3）が乗ぜられ、その積が最後に一定値に加算されて音声出力信号が発生されている。

【0061】前述した命令のマッピング及び順序付けステップ（図7のステップ350）では、データベース内の命令数を種々の方法で減少させている。

a) 2以上の論理的に隣接する命令が同じタイプの単一の命令で置換えられる場合は、この置換を行う。例えば、2つの論理的に隣接するビットシフト命令は、別々の命令内で言及した量の和だけシフトする単一のシフト命令に置換えることができる。

【0062】b) 2以上の論理的に隣接する命令が単一の異なる命令で置換えられる場合も、この置換を行う。例えば、（本例における如く）処理ユニットが単一クロック周期内に「乗加算」命令を行う能力があれば、論理的に隣接する乗算及び加算命令は、結合して単一の乗加算命令とする。ビットシフト命令を更にこれと併合して複合乗加算シフト命令とすることができる。

【0063】この減少方法を図9に示す。同図は、図8の部分で命令減少処理したものを示す。第1音声信号（音声1）に施される連続的乗算及び加算を結合して、単一の乗加算命令500としている。同様に、係数3による乗算及び一定値の加算を結合して、単一の乗加算命令510としている。

【0064】図10は、変数を遅延させたものを用いて処理する方法を示す模式図である。この技法は、特定の変数を発生して記憶し、それからあとで信号処理連鎖において当該変数の以前の値に対して操作を加える場合に、適用する。

【0065】本例では、図7のステップ350で、このタイプの状態が探索される。この状態が発生すると、前述の2つの動作（即ち、変数の発生及び変数の遅延値のあとでの使用）の実行命令を逆にし、遅延された変数を必要とする動作を実際には、当該変数を発生する前に行わせる。こうすると、その度毎に、該変数の以前に発生されたものが使用されることになる。

【0066】この技法を図10に模式的に示す。同図は、音声サンプル期間n及び次の音声サンプル期間n+



1の一部において処理ユニットp1により行われる処理を示す。命令520で、所要の変数の前回の値が一時的メモリ530から読出される。該変数の新しい値は、それから命令540で発生され、メモリ530に記憶され、あとで音声サンプル期間n+1内の対応する命令520'で読出される。

【0067】これに対し、図11は、以前提案された信号処理装置がこの問題をどのように処理したかを示す模式図である。図11において、変数は、命令540で発生され第1のメモリ(A)550に記憶される。該変数の遅延されたものがそれから、命令560で第2のメモリ(B)570から読出される。該変数は、命令555で第1メモリAから第2メモリBに転送される。

【0068】換言すると、以前提案された信号処理装置では、変数の発生及び使用を逆の順序で行う本例に比べ、2倍のメモリを必要とし、また、変数をメモリ間で転送する余分の命令555をも必要とした。

【0069】

【発明の効果】以上説明したとおり、本発明によれば、複数の並列処理ユニットをもつデジタル音声処理装置において異なる処理ユニットに効率よく処理タスクを配分することができる。

【図面の簡単な説明】

【図1】本発明を用いるデジタル音声ミキシング操作

卓の模式図である。

【図2】図1の操作卓において1音声チャンネルに加えられるチャンネル処理の一部を示す模式図である。

【図3】図1の信号プロセッサを形成する信号処理アレイの模式図である。

【図4】図3の処理アレイの動作の例1を示す模式図である。

【図5】図3の処理アレイの動作の例2を示す模式図である。

【図6】図1の操作卓用の目的コードの作成を示す模式フローチャートである。

【図7】図3の処理アレイ用の目的コードの作成及び配分を示す模式フローチャートである。

【図8】図6のネットリストの一部の例を示すブロック図である。

【図9】命令減少処理後の図8の部分を示すブロック図である。

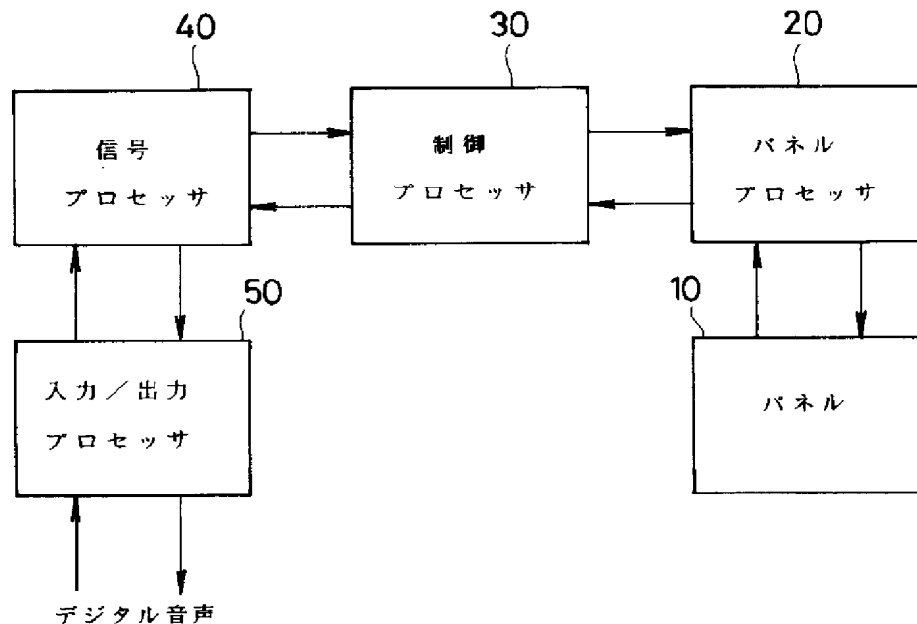
【図10】変数の遅延されたものを用いる信号処理方法の例を示す模式図である。

【図11】変数の遅延されたものを用いる従来の信号処理方法を示す模式図である。

【符号の説明】

p1～p(m+3) 並列処理ユニット

【図1】



本発明を用いるデジタル音声ミキシング操作卓

【図2】

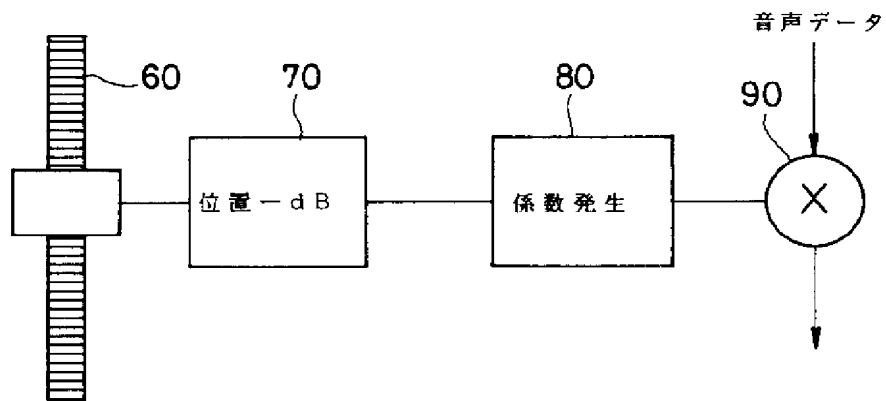


図1の操作卓におけるチャンネル処理の一部

【図3】

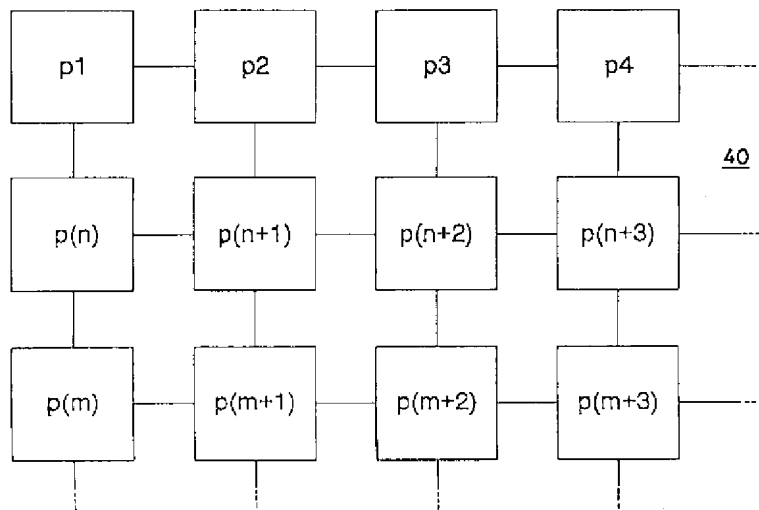


図1の信号プロセッサを形成する信号処理アレイ

【図4】

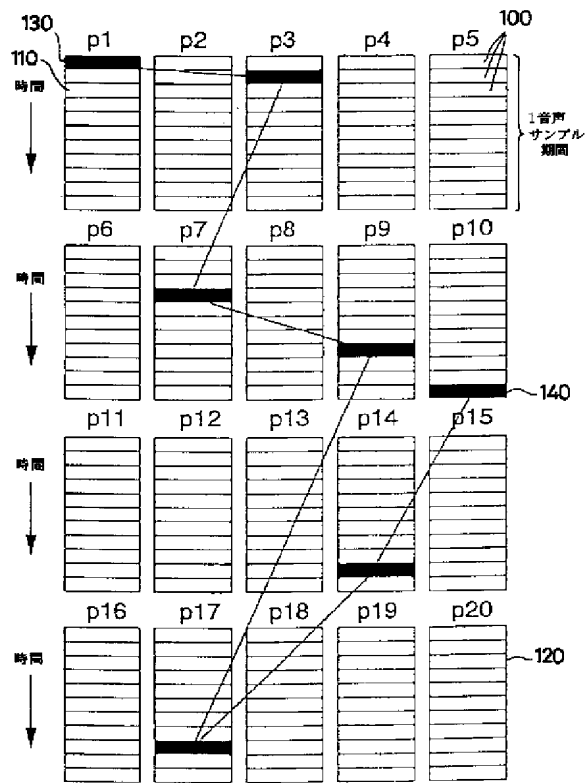


図3の処理アレイの動作例1

【図5】

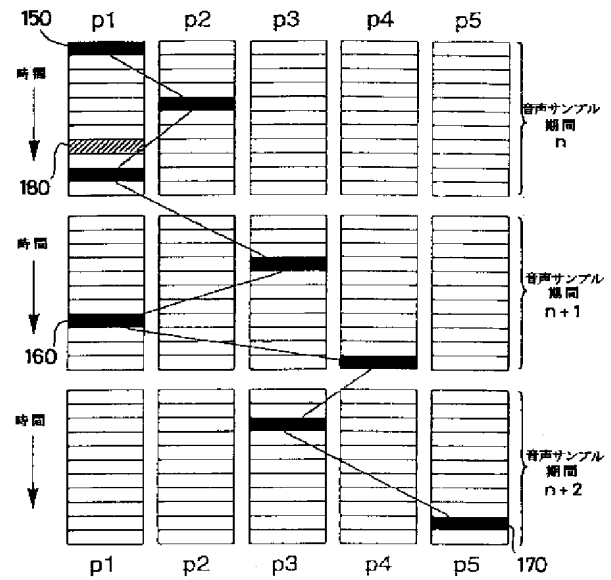


図3の処理アレイの動作例2

【图8】

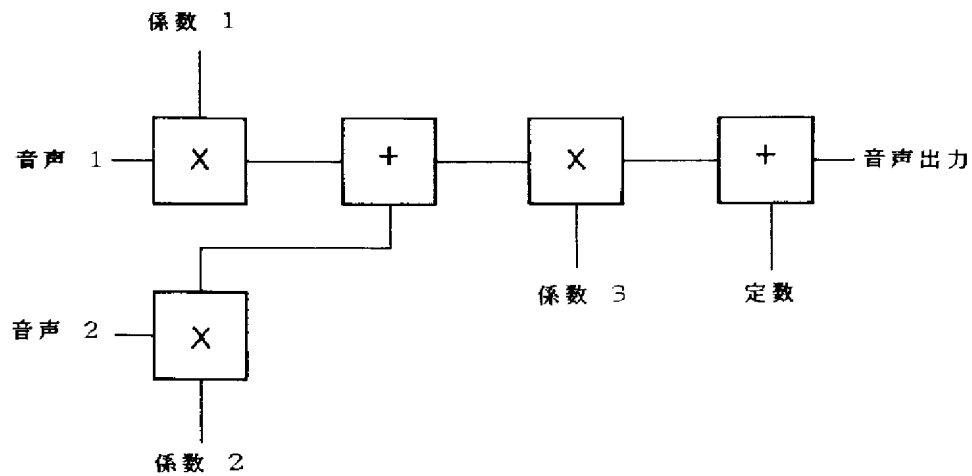


図6のネットリストの一部

【図6】

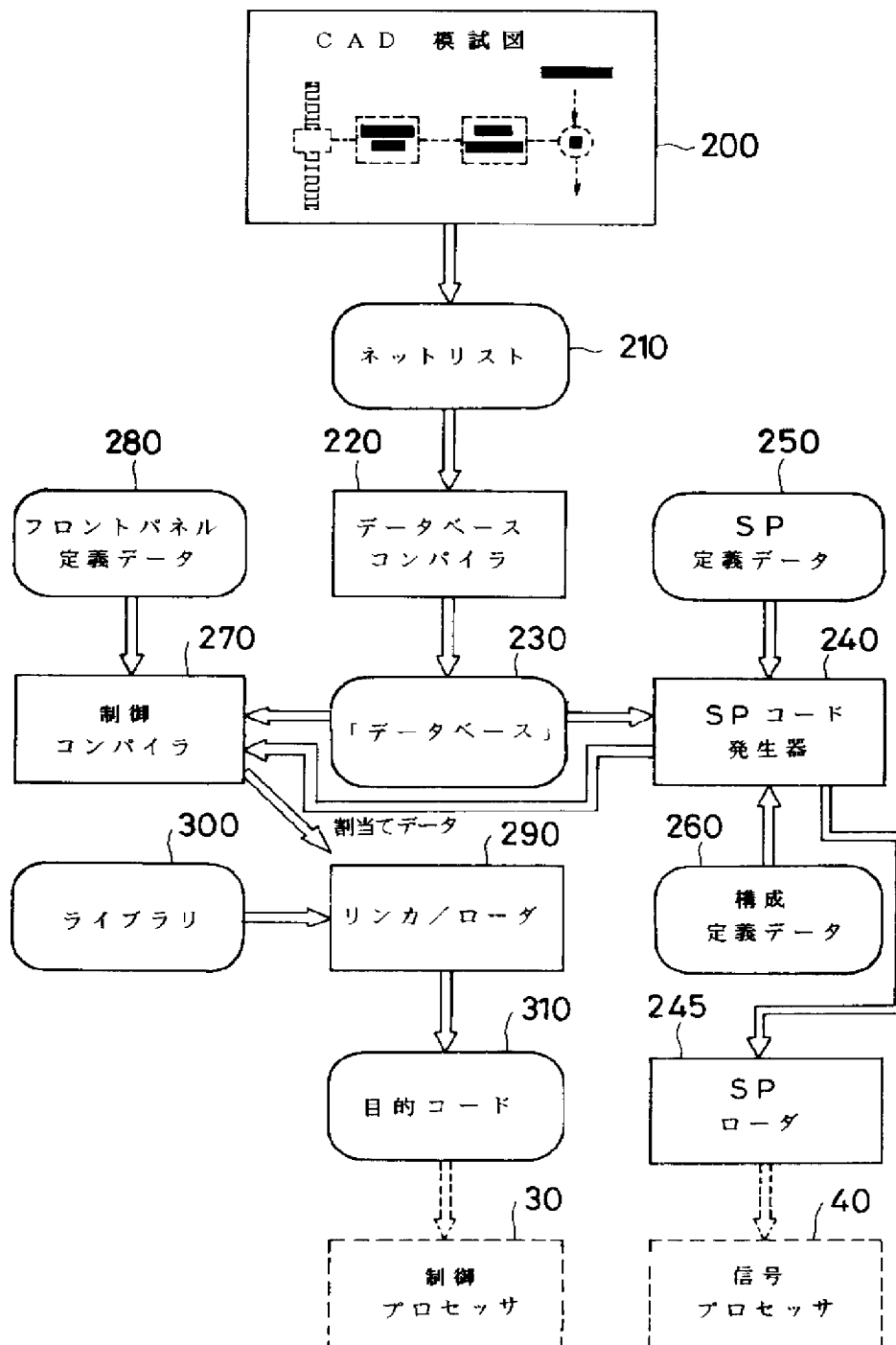


図1の操作卓内の全デバイス用目的コードの作成

【図7】

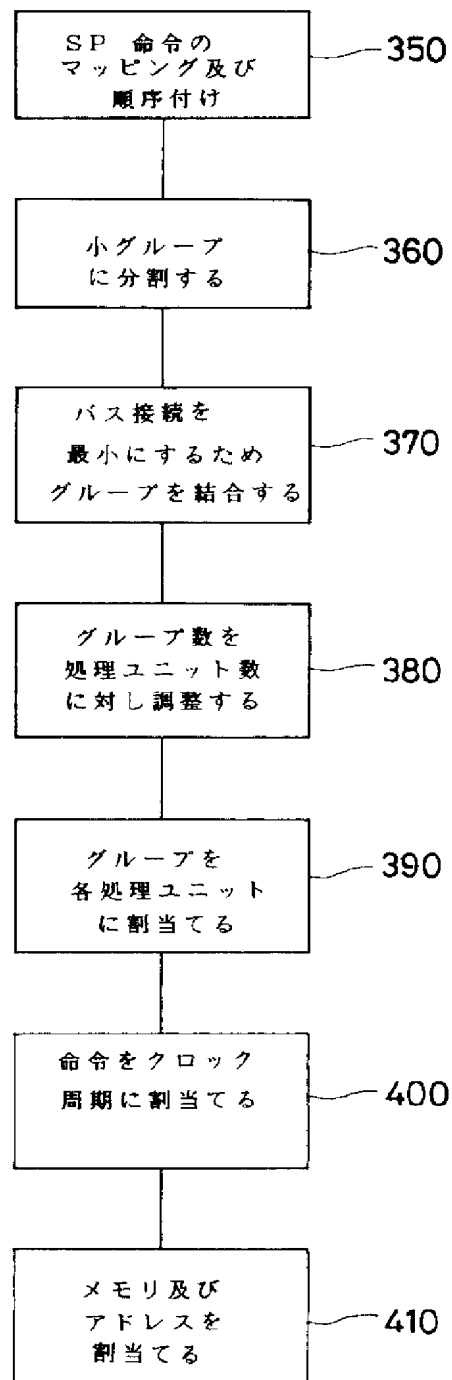


図3の処理アレイ用目的コードの作成と配分

【図9】

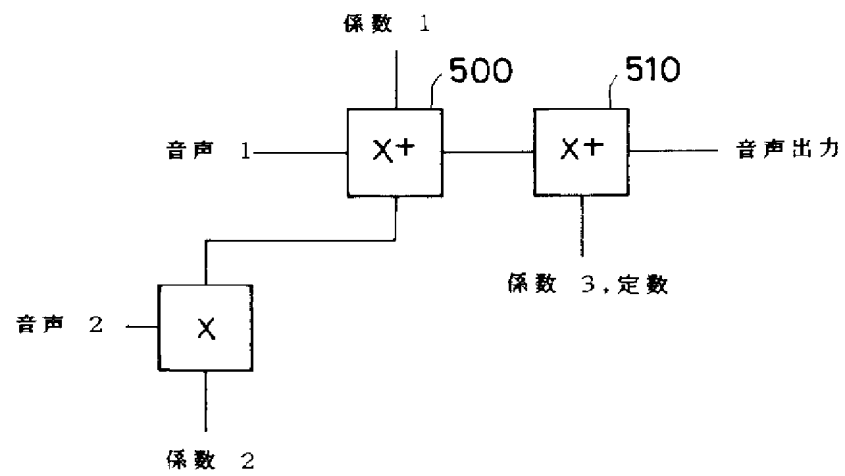
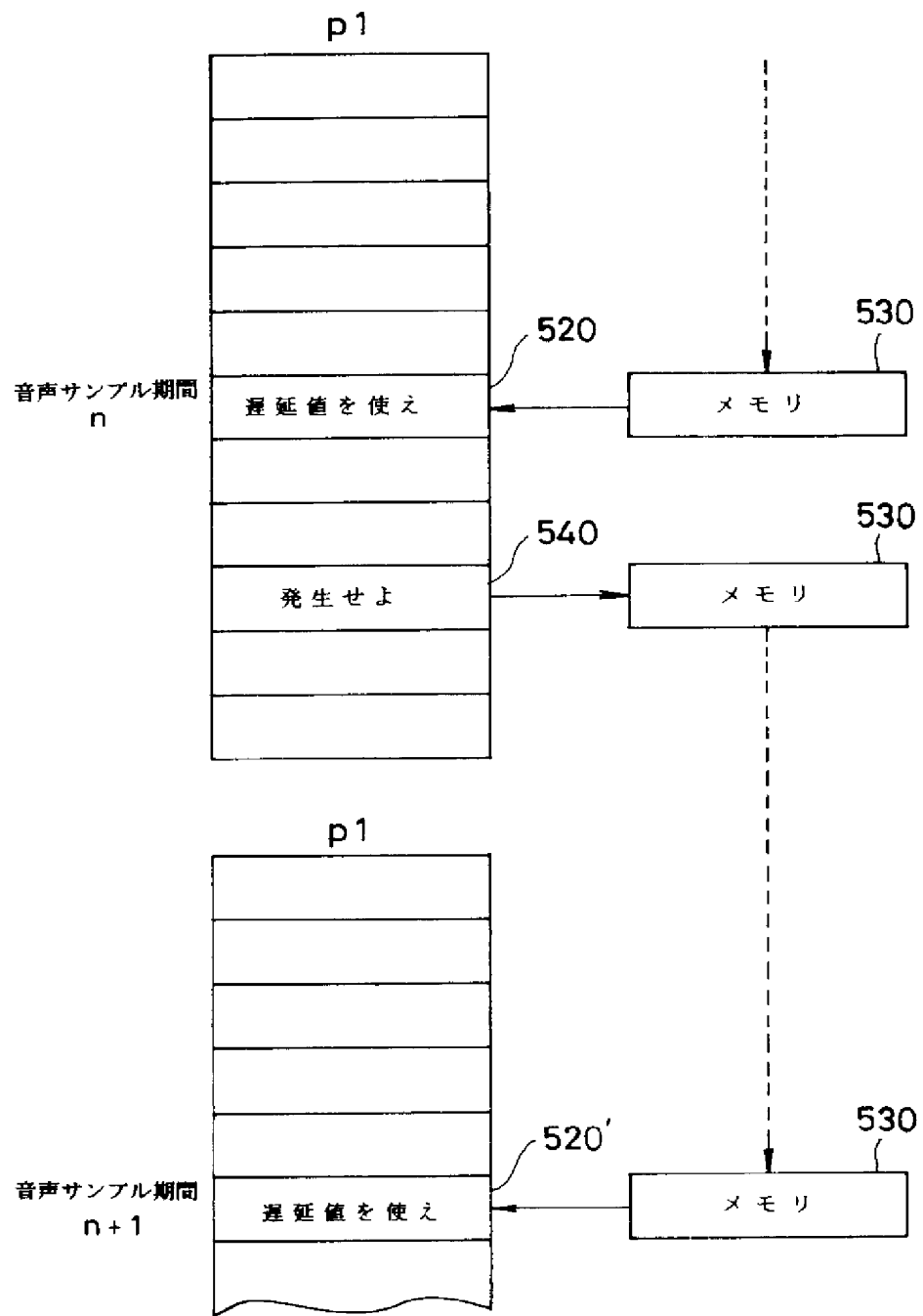


図8の命令を減らした例

【図10】



変数を遅延させて用いる処理方法の例

【図11】

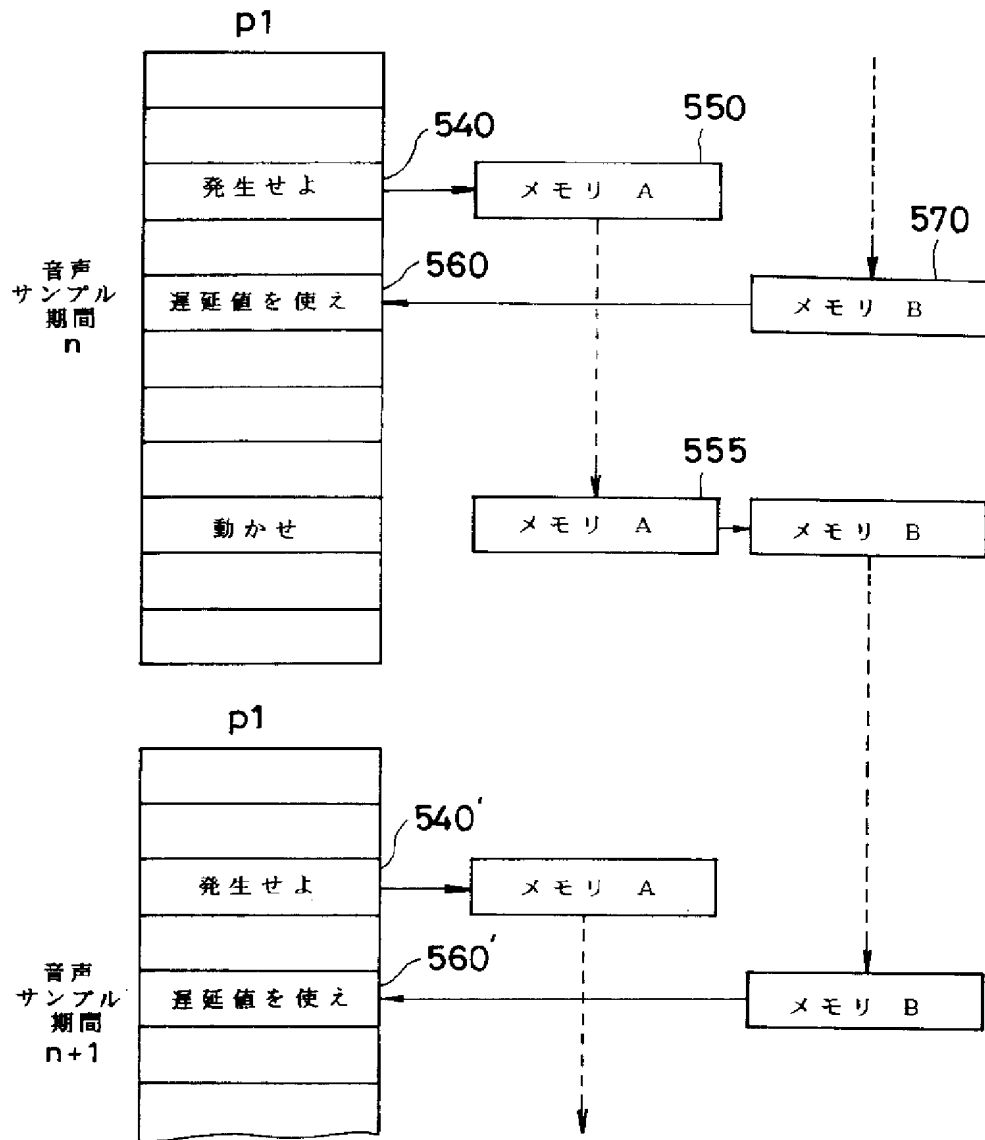


図10に対応する従来の方法

フロントページの続き

(72)発明者 ビーター チャールズ イースティ  
イギリス国 オックスフォード、フェアエ  
ーカーズ ロード 18

(72)発明者 ウィリアム エドマンド クランスタン  
ケンティッシュ  
イギリス国 オックスフォードシャー、チ  
ッピング ノートン、リーズ アプローチ